

1° Corso Introduttivo a GNU/Linux
Terza lezione

9 Giugno 2001

LUG Roma3

L'ambiente grafico di Linux

9 Giugno 2001

LUG Roma3 Stefano Francesi

Riferimenti bibliografici:

Appunti Linux

<http://www.pluto.linux.it/ildp/AppuntiLinux>

X.Org

<http://www.x.org>

Da Windows a Linux

<http://www.attivissimo.net>

MAN di BASH

X

X Window System è il sistema grafico per gli ambienti Unix (ANSI/POSIX). Sviluppato nei laboratori del MIT (*Massachusetts Institute of Technology*) intorno al 1986 è ora mantenuto da The Open Group / X Consortium. X si occupa della gestione di un ambiente grafico completo, compresi i vari dispositivi di input e puntamento ad esso indispensabili.

XFree86

Dalla versione 11R6.4 *X Window* non è più software libero. Lo sviluppo di *X* come software libero avviene per opera di **The XFree86 Project**, per il quale vale ancora la vecchia licenza MIT; il progetto è nato su piattaforme Intel x86 (da cui il nome) ma è ora disponibile anche per architetture Alpha, PowerPC e Sparc.

Altre ditte, invece, forniscono versioni commerciali di *X* dalle prestazioni spesso superiori a quelle di *XFree86*.

Struttura di X

Identifichiamo le parti fondamentali del sistema *X Window*:

- servente X
- cliente X
- protocollo X
- librerie Xlib
- gestore di finestre

Servente X

Il servente X è il programma che gestisce le funzionalità grafiche e le mette a disposizione degli altri programmi. Il servente grafico fornisce anche un servizio di rete dal momento che consente l'accesso a programmi in funzione presso altri elaboratori.

XFree86 fornisce un servente gratuito per ogni gruppo di schede video esistente con frequenti aggiornamenti.

Cliente X

I clienti X sono i programmi che utilizzano l'ambiente grafico comunicando con il servente X. Un cliente X può essere eseguito anche su un elaboratore diverso da quello sul quale è in funzione un servente X.

Protocollo X

Tra i clienti X e il servente X, intercorre una comunicazione, attraverso un protocollo prestabilito, in grado di mascherare le peculiarità del sistema operativo e l'hardware sottostante. Questo fornisce l'elevata portabilità sia dei serveri che dei clienti, senza penalizzarne le prestazioni.

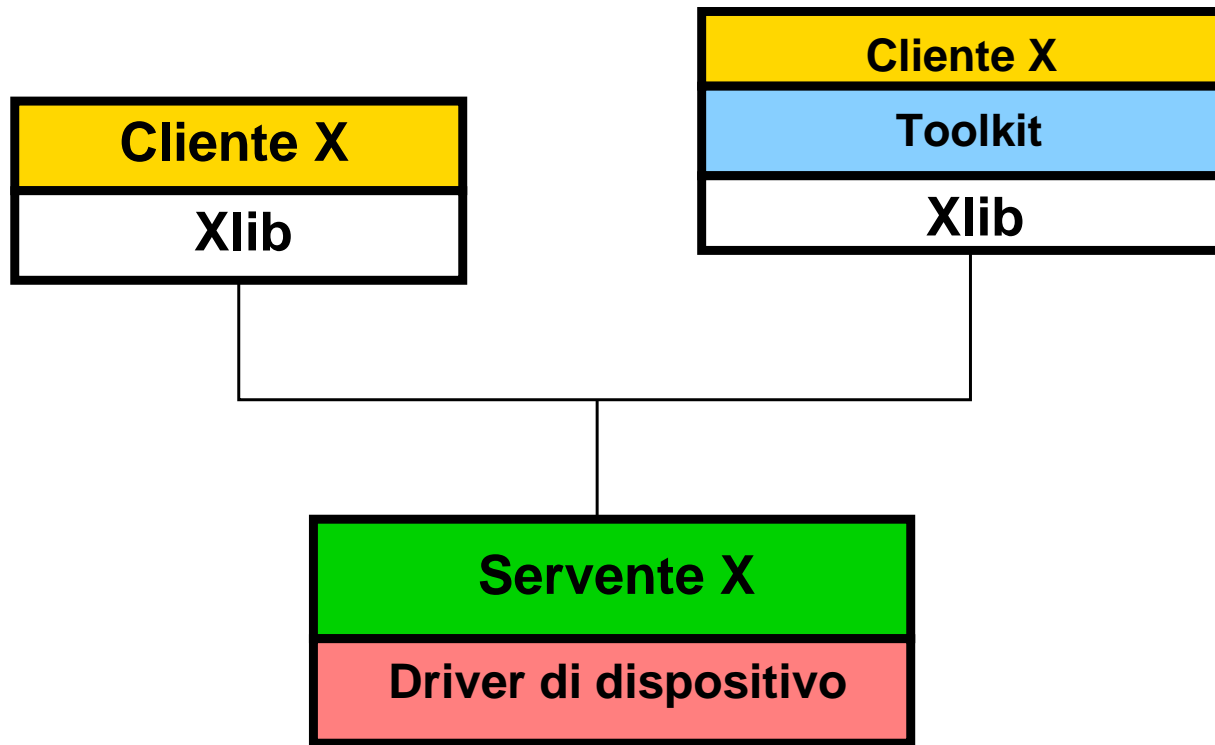


Figure 1: Protocollo X

Librerie Xlib

I programmi che utilizzano i servizi del server X utilizzano funzioni di libreria specifiche, conosciute come Xlib, indispensabili anche per i clienti che utilizzano servizi di server remoti.

Gestore di finestre

Un gestore di finestre (altrimenti detto *window manager*) è un programma speciale che si occupa di gestire l'aspetto delle finestre delle varie applicazioni. In genere un *window manager* equivale ad un cliente X.

I moderni gestori di finestre

Esistono svariati *window manager* e *desktop environment* già forniti insieme alla propria distribuzione o liberamente scaricabili dalla rete.

Ne elenchiamo qui soltanto i più usati:

- Gnome
- KDE
- Enlightenment
- Sawfish
- XFce
- WindowMaker
- Fvwm

GNOME

GNOME (*GNU Network Object Model Environment*) è un *desktop environment* completamente open source, di semplice installazione ed offre un potente ambiente di programmazione per gli sviluppatori.



GNOME (2)

GNOME non ha un proprio *window manager*: si appoggia su altri quali Enlightenment e Sawfish. Offre invece ai programmi X una libreria molto potente e versatile chiamata *GTK+* tramite la quale è possibile costruire l'interfaccia grafica del proprio programma Gnome. È ovviamente sufficiente avere installate le librerie di Gnome per utilizzare programmi Gnome, pur utilizzando altri window manager quali KDE o WindowMaker.

KDE

KDE (*K Desktop Environment*) è come Gnome un *desktop environment* ma è fornito di un suo *window manager* dallo stile veramente molto simile a MS-Windows. È basato su un'ottima libreria grafica (chiamata *QT*), che tuttavia non viene rilasciata sotto licenza GPL e per questo è bandita dalle distribuzioni puramente GNU quali Debian.



Enlightenment e Sawfish

Enlightenment e **Sawfish** sono due window managers normalmente legati all'ambiente GNOME: il primo è molto scenografico (trasparenze, animazioni, effetti sonori) e sta per subire pesanti innovazioni che lo porteranno ad assomigliare all'interfaccia *Aqua* di Apple MacOS X; il secondo è più semplice ma altamente configurabile.

XFce

XFce è *desktop environment* leggerissimo e completo con un caratteristico pannello che ricorda molto CDE (*Common Desktop Environment*), un DE commerciale che gira su parecchi UNIX. È basato come Gnome sulle librerie *GTK+*.



WindowMaker

WindowMaker è molto elegante e fortemente legato alle applicazioni GNU. Si ispira alla vecchia interfaccia NeXTSTEP del mondo UNIX.



Fvwm, Fvwm2, Fvwm95

Sono window manager molto vecchi, piuttosto minimali, i primi a tentare un avvicinamento all'aspetto del sistema MS-Windows.

Collocazione nel file system

La struttura prevista per il file system di GNU/Linux colloca tutti i file di X (binari, documentazione, librerie, ecc.) sotto `/usr/X11R6`.

I file di configurazione, invece, si trovano sotto `/etc/X11`.

Per ragioni di compatibilità, vengono aggiunti alcuni **collegamenti simbolici**.

```
/usr/bin/X11      -> /usr/X11R6/bin
/usr/lib/X11      -> /usr/X11R6/lib/X11
/usr/include/X11 -> /usr/X11R6/include/X11
```

Collegamenti

Una tecnica molto usata nei sistemi Linux è quella dei collegamenti tra file.

Si distinguono due tipi:

- collegamenti simbolici (*symbolic link*)

Sono file speciali che **si distinguono** dagli originali e aiutano ad evitare la duplicazione dei file per motivi di compatibilità. Sono anche utilissimi per semplificare i path di esecuzione.

- collegamenti fisici (*hard link*)

Creano riferimenti a dati esistenti in maniera **non distinguibile** dagli originali.

Si creano con il comando `ln`.

Collegamenti simbolici

Sono file speciali distinguibili dagli originali. Si creano con il comando `ln`, con l'opzione `-s`. Ad es.

```
bash$ touch file_prova
bash$ ln -s file_prova simbolico
bash$ ls -l
```

```
-rw-r--r--  1 utente1  corsi    33 May 31 11:23 file_prova
lrwxr-xr-x  1 utente1  corsi    10 May 31 11:24 simbolico -> file_prova
```

La lettera `l` che appare all'inizio dei permessi indica esplicitamente che si tratta di un link simbolico.

Gli altri permessi non sono significativi dal momento che contano solamente i permessi del file originale.

Collegamenti fisici

Vanno usati solo in situazioni di particolari necessità: creano riferimenti a dati esistenti in maniera non distinguibile da quelli originali.

Quando si cancella un file con collegamenti fisici esso sarà rimosso solamente quando saranno stati cancellati tutti i riferimenti a questo.

```
bash$ echo "prova" >primo_file
bash$ ln primo_file secondo_file
bash$ ls -l
```

```
-rw-r--r--  2 utentel  corsi          6 May 31 12:05 primo_file
-rw-r--r--  2 utentel  corsi          6 May 31 12:05 secondo_file
```

I due file sono effettivamente **lo stesso file** dunque ogni modifica su uno dei due si riscontrerà ovviamente anche sull'altro.

Configurazione di XFree86

Le varie distribuzioni offrono programmi testuali e grafici per la configurazione dell'ambiente grafico.

- RedHat `xconfigurator`
- Mandrake `XFdrake`, `DrakX`
- Suse `sax`, `sax2`

L'unica certezza in caso di funzionamento non corretto di X è il file `/etc/X11/XF86Config`.

La sua complessa struttura esula dagli scopi di questo corso.

Avvio di X

Il server X è un programma e può essere semplicemente avviato dalla shell digitando X.

```
bash$ X
```

Avvio di X (2)

La schermata che ci si trova davanti è completamente grigia con un cursore nero: questo è un server grafico. Manca ora un *gestore di finestre* e soprattutto l'indispensabile terminale. Per uscire da X è necessario premere la combinazione di tasti Ctrl+Alt+Backspace.

Avvio di X (3)

La maniera corretta di avviare l'ambiente grafico di Linux è digitando `startx`. In questo modo si avvia il server X ed il gestore di finestre di default della propria configurazione. Il file di configurazione locale che specifica il *window manager* da avviare è in genere `.xinitrc` ed è contenuto nella propria home. Ogni *window manager* ha programmi di utilità in grado di generare automaticamente questo file.

xinit

In realtà `startx` è soltanto uno script di shell che avvia un programma chiamato `xinit`. È `xinit` infatti ad occuparsi dell'avvio di un server X e di un primo programma cliente in grado di utilizzarlo (in genere il terminale).

```
xinit [cliente] [ -- [server] [stazione_grafica] ]
```

startx equivale a:

- `xinit ~/.xinitrc -- /usr/X11R6/bin/X :0.0`
- `xinit xterm -- /usr/X11R6/bin/X :0.0`
(se il file non esiste)

xinit (stazione grafica)

Cosa vuol dire il parametro `:0.0`? Poichè *X* si basa su un'architettura client/server, è possibile avviare più server e più clienti sulla stessa macchina. La stazione grafica (o *display*) è dunque identificata da `numero[.schermo]`.

Il primo server sarà richiamabile sulla prima console virtuale libera, che solitamente è la settima (`Ctrl+Alt+F7`), i successivi con `Ctrl+Alt+F8`, `Ctrl+Alt+F9`, e così via.

Se il primo server ha a disposizione più schermi fisici essi saranno identificati con `0.0`, `0.1`, `0.2` e così via.

Avvio di più server X

- [CTRL-ALT-F1] (per passare alla prima console)
- inserire nome e password (se necessario)
- startx [INVIO]
(per far partire il server su :0)
- [CTRL-ALT-F2] (per passare alla seconda console)
- inserire nome e password (se necessario)
- startx -- :1 [INVIO]
(per far partire il server su :1)

Avvio di window manager diversi

- `startx [INVIO]`
(per far partire il server su `:0`)
- `[CTRL-ALT-F2]` (per passare alla seconda console)
- inserire nome e password (se necessario)
- `xinit /usr/X11R6/bin/wmaker -- :1 [INVIO]`
(per far partire il server su `:1`)
- `[CTRL-ALT-F7]` (per passare al primo server)
- `[CTRL-ALT-F8]` (per passare al secondo server)

Primi programmi per X

- `xterm` Terminale
- `xcalc` Calcolatrice scientifica
- `xclock` Orologio analogico
- `xv` Visualizzatore/conversore immagini bitmapped
- `xeyes` Due occhi che seguono il puntatore del mouse

Ampersand (&)

Utilizzando il terminale in modalità grafica è ovviamente possibile avviare più programmi contemporaneamente. Per far questo è necessario utilizzare il simbolo & (*ampersand* o *E commerciale*).

Il terminale risponderà con [numero_job] PID_processo

Ad es.

```
[utente1@dia1 utente1]$ xclock &  
[3] 25691
```

Controllo dei job

Linux permette di sospendere e ripristinare selettivamente l'esecuzione dei processi. La shell associa un *job* ad ogni processo avviato da terminale.

Ci sono due tipi di job:

- in primo piano o in *foreground*;
- sullo sfondo, o asincroni, o in *background*.

Comandi sui job

- `jobs` Mostra la lista dei job *sospesi* o in *background* nella shell corrente;
- `CTRL-Z` Sospende un job in *foreground*
- `CTRL-C` Termina un job in *foreground*
- `bg` Pone l'ultimo job sospeso in *background*
- `fg` Pone l'ultimo job sospeso in *foreground*
- `nomecomando &` Avvia `nomecomando` in *background*
- `kill %i` Uccide l'*i*-esimo job

nohup

Il comando `nohup` permette un'estensione ulteriore al meccanismo di esecuzione in background dei processi. Qualsiasi processo avviato da una shell, compresi quelli in background, viene terminato quando la shell viene chiusa.

Eseguendo invece `nohup nomecomando &` il processo continuerà a sopravvivere anche quando l'utente non sarà più loggato al sistema.

L'output del comando sarà redirezionato sul file `nohup.out`

La variabile d'ambiente DISPLAY

La variabile d'ambiente DISPLAY indica ai clienti X quale server X dovranno utilizzare per il loro output grafico. La sintassi estesa è:

```
DISPLAY = [nomehost]:numero.[schermo]
```

Esempio con due server X

Sul terminale del server :0

```
bash
```

```
export DISPLAY=:1
```

```
xeyes &
```

Sul terminale del server :1

```
bash
```

```
export DISPLAY=:0
```

```
xcalc &
```

[Ctrl+Alt+F7] e [Ctrl+Alt+F8] per spostarsi tra i due server.

xhost

Chiunque abbia accesso ad un dato host può avviare a piacimento clienti X. Per abilitare/disabilitare la richiesta da parte di alcuni host si utilizza il comando `xhost`.

Ad es.

`xhost +dia1` Abilita l'host `dia1` ad accedere al proprio servente X.

`xhost -dia3` Impedisce all'host `dia3` di accedere al servente X.

`xhost +` Abilita l'accesso da tutti gli host.

`xhost -` Impedisce l'accesso da qualsiasi host.

xdm

XDM (*X Display Manager*) è la versione grafica del prompt di login e si occupa di avviare le sessioni X degli utenti. Fornisce tra l'altro l'**XDMCP** (*X Display Manager Control Protocol*), un meccanismo per richiedere una sessione X da un host remoto. Avendo a disposizione **xdm** già avviato su una macchina è possibile utilizzare il proprio server X per effettuare un login grafico sulla macchina che lo fornisce come se ci si fosse effettivamente seduti davanti a essa. È anche possibile in tal modo realizzare terminali *diskless* ovvero dotati di solo processore e monitor e permettere comunque l'utilizzo dell'interfaccia grafica.

I *desktop environment* tendono spesso a fornire anche la loro versione di **xdm**, ad es. **gdm** per Gnome e **kdm** per KDE.

Utilizzo di XDMCP

X -query nomehost Richiede a nomehost il login grafico utilizzando il suo xdm.

X -broadcast Fa la richiesta precedente in *broadcast* ovvero a tutti i computer della propria rete locale.

X -indirect nomehost Avvia il *chooser* di nomehost ovvero un piccolo menu grafico che permette di scegliere tra i computer della propria rete locale dotati di xdm.

GLX

Negli ultimi due anni anche Linux si è avvicinato seriamente alla grafica 3D e gli sviluppatori di driver per le schede video si sono chiesti se fosse possibile ottenerne versioni *accelerate*.

Matrox ed altri produttori hardware hanno avviato un progetto di ricerca che è culminato nel modulo **GLX**, che permette alla grafica OpenGL di sfruttare l'accelerazione 3D della propria scheda video.

Il progetto è stato poi riassorbito da XFree86 con l'uscita della versione 4.0

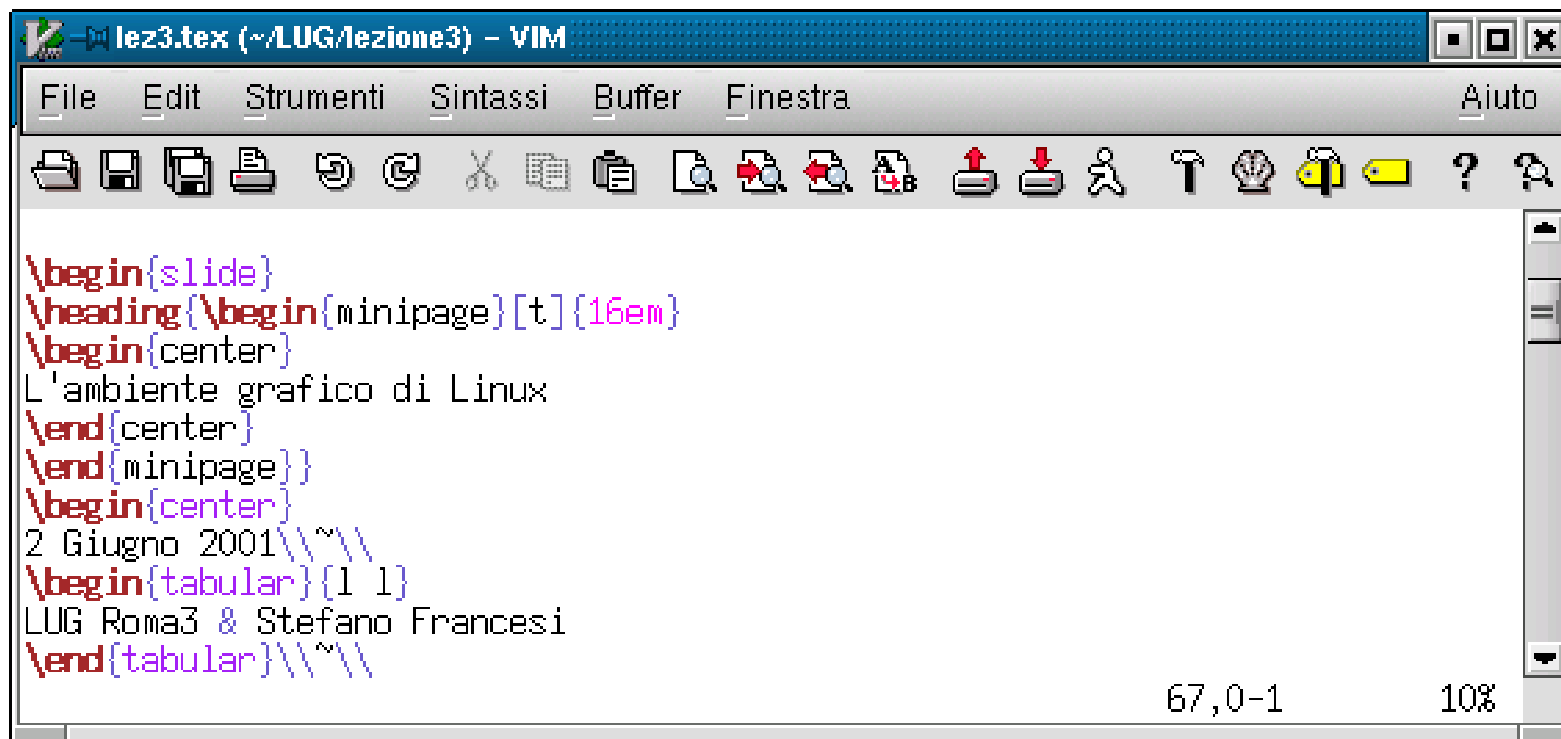
Programmi per X

I programmi che sfruttano l'interfaccia grafica di Linux sono innumerevoli. I programmi inizialmente sviluppati da GNU per convenzione iniziano tutti per x.

I due window manager più in voga, Gnome e KDE, hanno esteso la convenzione: la maggior parte dei programmi Gnome iniziano per g, mentre quelli per KDE iniziano per k.

Non è ovviamente sempre vero il viceversa.

Vediamo di seguito alcuni screenshot dei programmi di utilità più usati.



The image shows a screenshot of the gvim editor window. The title bar reads "lez3.tex (~/.LUG/lezione3) - VIM". The menu bar includes "File", "Edit", "Strumenti", "Sintassi", "Buffer", "Finestra", and "Aiuto". The toolbar contains various icons for file operations and editing. The main text area displays LaTeX code for a slide, with some commands highlighted in red and purple. The code is as follows:

```
\begin{slide}
\heading{\begin{minipage}[t]{16em}
\begin{center}
L'ambiente grafico di Linux
\end{center}
\end{minipage}}
\begin{center}
2 Giugno 2001\\~\\
\begin{tabular}{1 1}
LUG Roma3 & Stefano Francesi
\end{tabular}\\~\\

```

The status bar at the bottom right shows "67,0-1" and "10%".

Figure 2: **gvim**: Vi IMproved per X

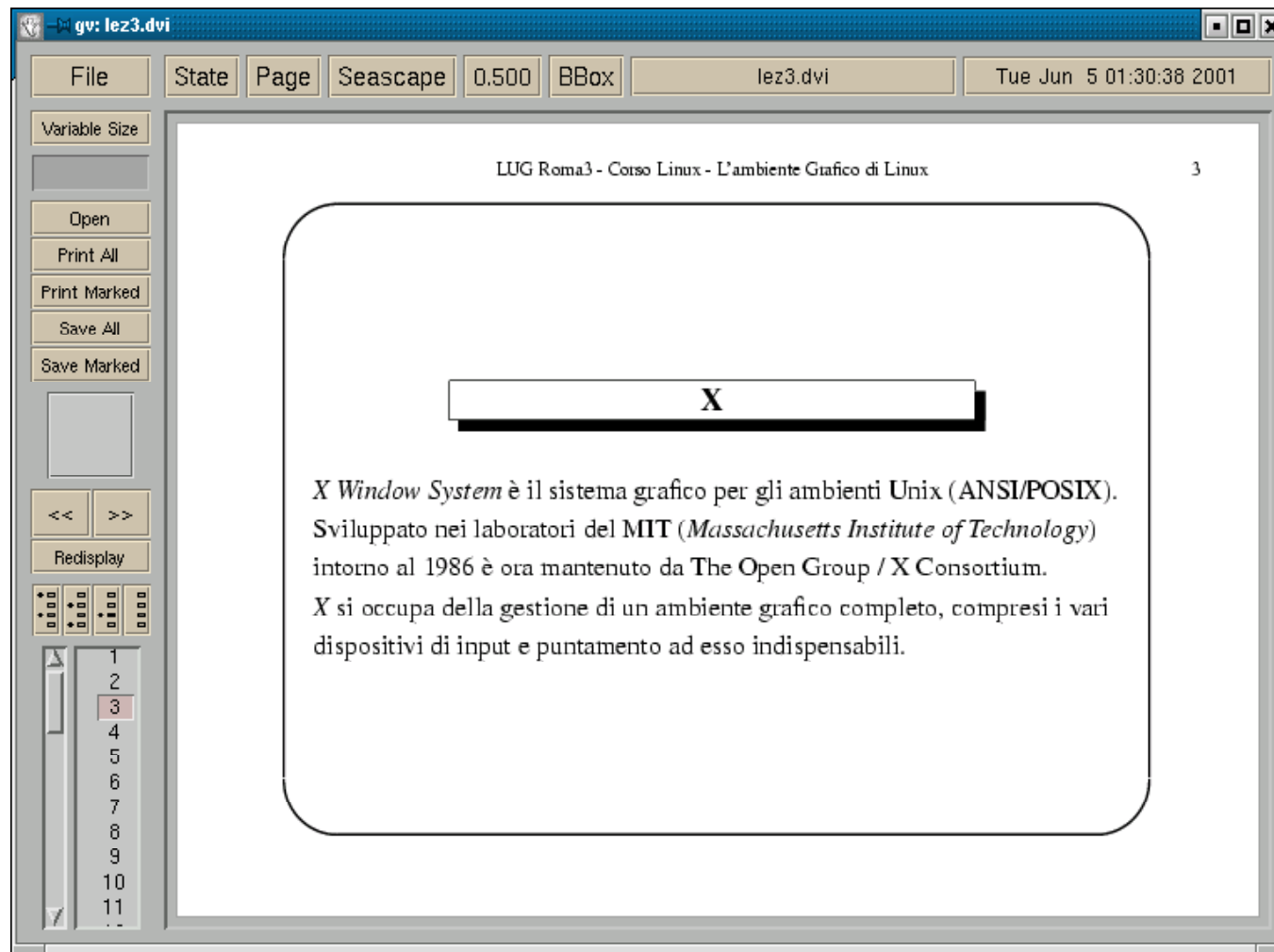


Figure 3: **gv**: Visualizzatore Postscript

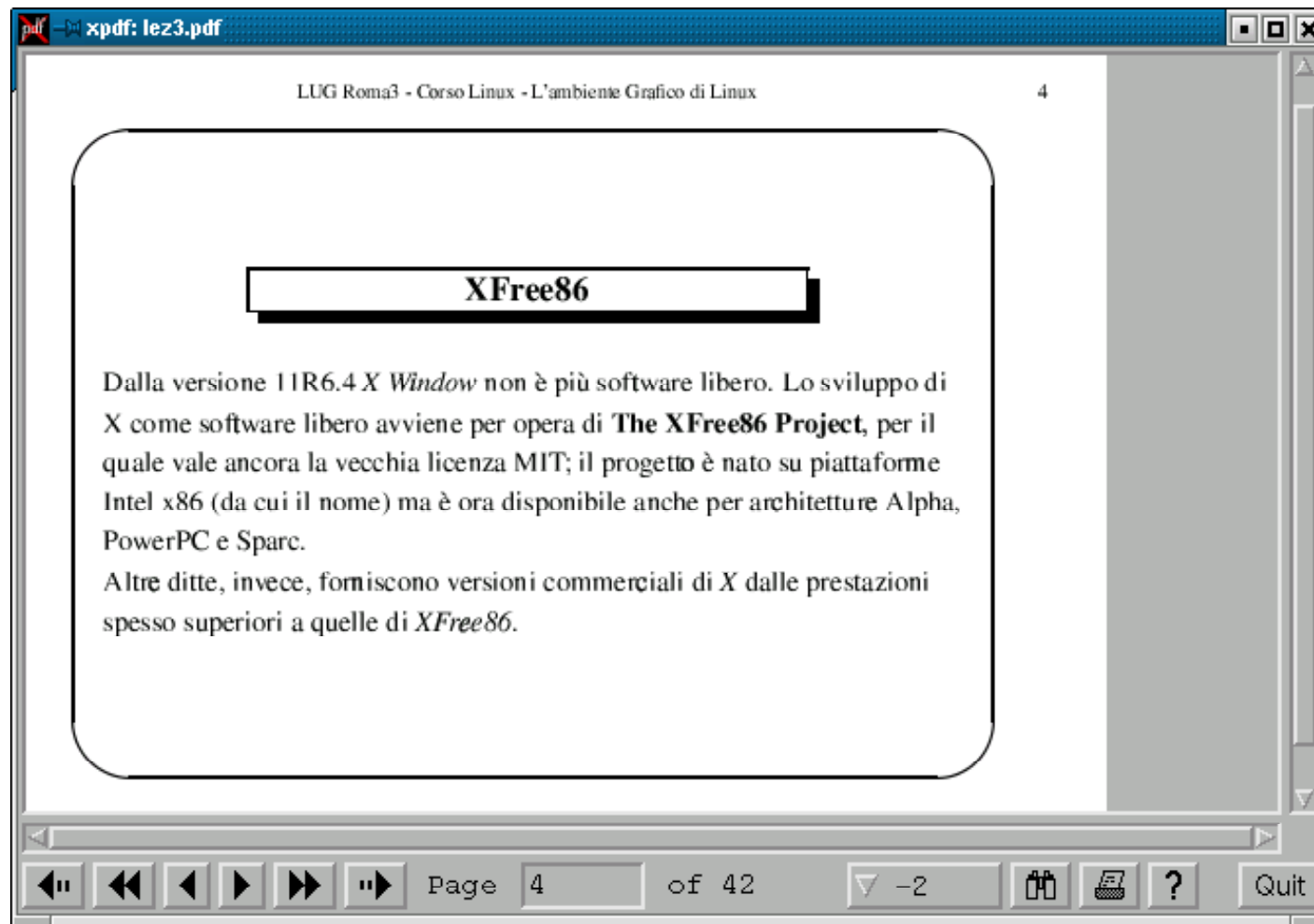


Figure 4: **xpdf**: Visualizzatore PDF

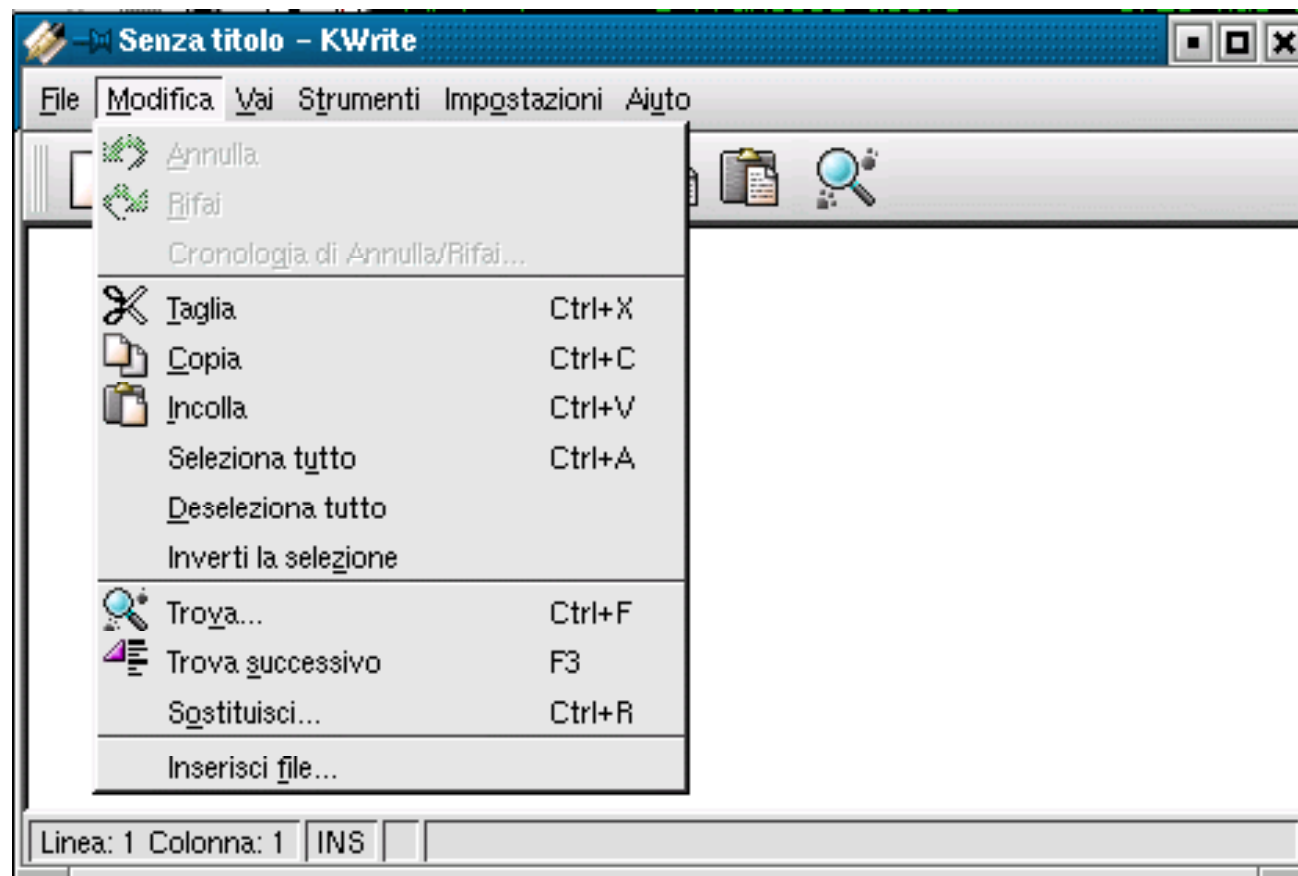


Figure 5: **kwrite**: Semplice editor testi di KDE

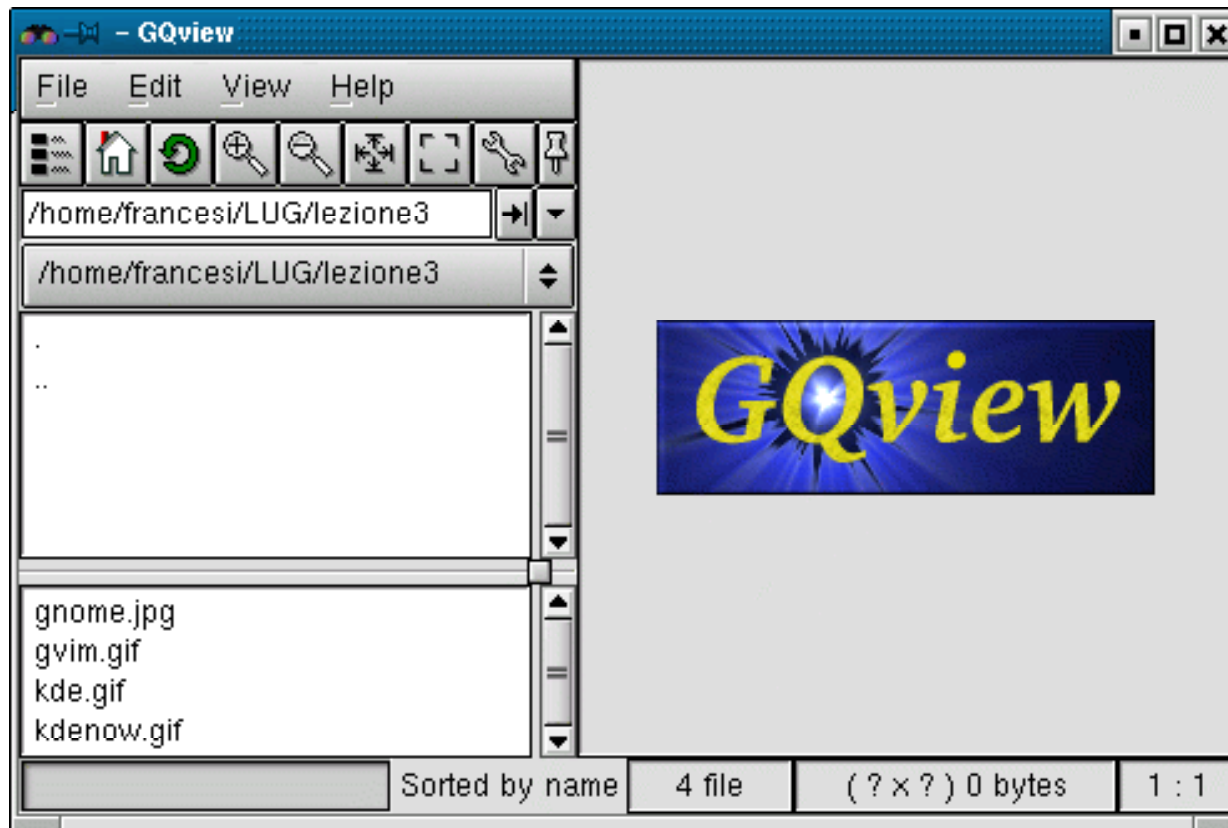


Figure 6: **gqview**: Slideshow di immagini di Gnome



Figure 7: **xmms**: Player MP3

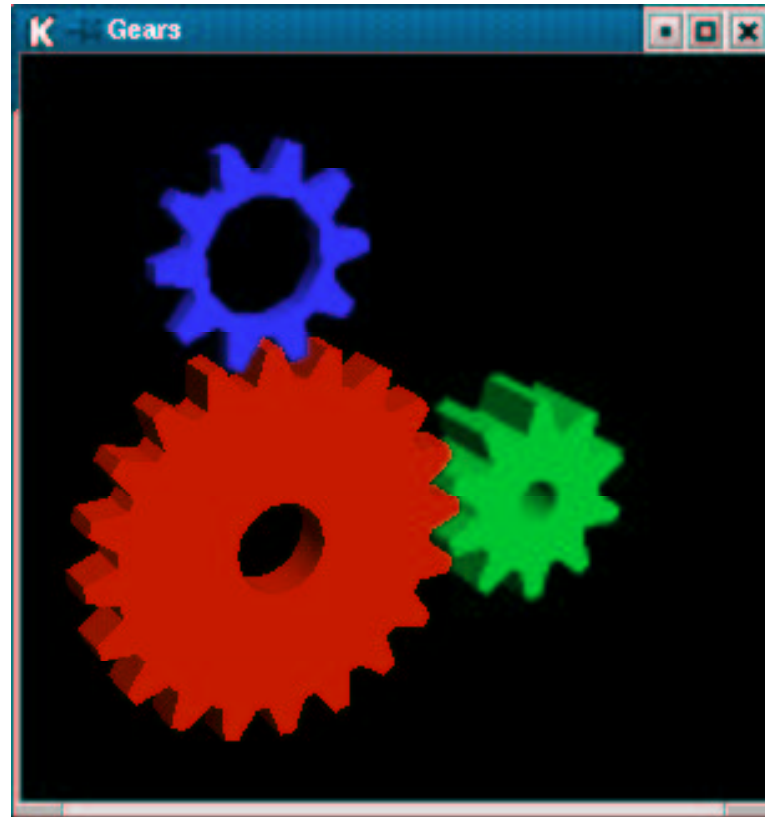


Figure 8: **gears**: Ingranaggi rotanti 3D